

Listing of the Claims:

Please cancel claims 7, 10, 17, 26, and 29-36, replace claims 1, 11, and 20, and add new claims 37-45, all as shown below.

1. (Currently Amended): A method for code completion, comprising:
providing a representation of a first program in a first programming language;
establishing a location in the first program;
associating the location with a representation of the first program;
obtaining code completion information relevant to the location in the first program based on the representation of the first program; and
wherein the obtaining occurs at the behest of an extensible compiler framework;
wherein the extensible compiler framework can integrate and interact with compilers for different programming languages through a common interface; and
wherein the extensible compiler framework provides code completion services to clients through a client interface.
2. (Original): The method of claim 1 wherein:
the location in the first program is one of: 1) a textual offset; 2) a structural navigation through a parse tree; 3) at least one semantic entity in the first program; and 4) a token or token range.
3. (Original): The method of claim 1 wherein:
the representation of the first program is a parse tree.

4. (Original): The method of claim 3 wherein:
the code completion information is based on information related to a node in the parse tree.
5. (Original): The method of claim 1 wherein:
the code completion information includes at least one of: 1) a class name and/or definition;
2) a type name and/or definition; 3) a field/member/variable name and/or definition; 4) a method
name and/or definition; and 5) a function name and/or definition.
6. (Original): The method of claim 1, further comprising:
analyzing the syntactic structure of a first program in a first programming language, wherein
the first program can be represented by a first set of tokens;
7. (Cancelled)
8. (Original): The method of claim 1 wherein:
the first program in the first programming language can be nested within a second program in
a second programming language.
9. (Original): The method of claim 1 wherein:
a second program in a second programming language is nested within the first program in the
first programming language.

10. (Cancelled)
11. (Currently Amended): A system for code completion, comprising:
- a component operable to provide a representation of a first program in a first programming language;
 - a component operable to establish a location in the first program;
 - a component operable to associate the location with a representation of the first program;
 - a component operable to obtain code completion information relevant to the location in the first program based on the representation of the first program; and
- wherein the obtaining occurs at the behest of an extensible compiler framework;
- wherein the extensible compiler framework can integrate and interact with compilers for different programming languages through a common interface; and
- wherein the extensible compiler framework provides code completion services to clients through a client interface.
12. (Original): The system of claim 11 wherein:
- the location in the first program is one of: 1) a textual offset; 2) a structural navigation through a parse tree; 3) at least one semantic entity in the first program; and 4) a token or token range.
13. (Original): The system of claim 11 wherein:
- the representation of the first program is a parse tree.

14. (Original): The system of claim 13 wherein:
the code completion information is based on information related to a node in the parse tree.
15. (Original): The system of claim 11 wherein:
the code completion information includes at least one of: 1) a class name and/or definition;
2) a type name and/or definition; 3) a field/member/variable name and/or definition; 4) a method
name and/or definition; and 5) a function name and/or definition.
16. (Original): The system of claim 11, further comprising:
a component operable to analyze the syntactic structure of a first program in a first
programming language, wherein the first program can be represented by a first set of tokens;
17. (Cancelled)
18. (Original): The system of claim 11 wherein:
the first program in the first programming language can be nested within a second program in
a second programming language.
19. (Original): The system of claim 11 wherein:
a second program in a second programming language is nested within the first program in the
first programming language.

20. (Currently Amended): A machine readable medium having instructions stored thereon that when executed by a processor cause a system to:

provide a representation of a first program in a first programming language;

establish a location in the first program;

associate the location with a representation of the first program;

obtain code completion information relevant to the location in the first program based on the representation of the first program; and

wherein the obtaining occurs at the behest of an extensible compiler framework;

wherein the extensible compiler framework can integrate and interact with compilers for different programming languages through a common interface; and

wherein the extensible compiler framework provides code completion services to clients through a client interface.

21. (Original): The machine readable medium of claim 20 wherein:

the location in the first program is one of: 1) a textual offset; 2) a structural navigation through a parse tree; 3) at least one semantic entity in the first program; and 4) a token or token range.

22. (Original): The machine readable medium of claim 20 wherein:

the representation of the first program is a parse tree.

23. (Original): The machine readable medium of claim 22 wherein:

the code completion information is based on information related to a node in the parse tree.

24. (Original): The machine readable medium of claim 20 wherein:

the code completion information includes at least one of: 1) a class name and/or definition; 2) a type name and/or definition; 3) a field/member/variable name and/or definition; 4) a method name and/or definition; and 5) a function name and/or definition.

25. (Original): The machine readable medium of claim 20, further comprising instructions that when executed cause the system to:

analyze the syntactic structure of a first program in a first programming language, wherein the first program can be represented by a first set of tokens;

26. (Cancelled)

27. (Original): The machine readable medium of claim 20 wherein:

the first program in the first programming language can be nested within a second program in a second programming language;

28. (Original): The machine readable medium of claim 20 wherein:

a second program in a second programming language is nested within the first program in the first programming language.

29 - 36. (Cancelled)

37. (New): The method of claim 1, wherein after a user enters an identifier name followed by a dot, code completion presents the user with a list of all possible valid names that could follow the dot.

38. (New): The method of claim 1, wherein after a user enters a method name, code completion presents the user with a template of arguments for a method, and if there is more than one method with the same name, code completion presents the programmer with a list of templates of arguments for all of the methods with the same name.

39. (New): The method of claim 1, wherein the client interface includes methods which accept arguments describing location within a source code file.

40. (New): The method of claim 39, wherein the extensible compiler framework can map a method to a language model associated with the source code file at a specified location.

41. (New): The method of claim 40, wherein the specified location is mapped to a location in a language model's parse tree.

42. (New): The method of claim 41, wherein information in the parse tree is used in conjunction with information in a name space to generate a list of possible code completions.

43. (New): The method of claim 42, wherein a client provides code completion to a user by presenting a pop-up window with a list of possible code completions.

44. (New): The method of claim 43, wherein as the user enters more text, the list of possible code completions shrinks to include only those completions that are still valid.

45. (New): The method of claim 43, wherein if the user selects a code completion, then the client inserts the code completion into the source code.